

Amendments to the Claims

1 Claim 1 (currently amended): A computer program product for enhancing performance of a  
2 multithreaded application, said computer program product embodied on a computer-readable  
3 medium and comprising:

4 computer-readable program code means for executing a plurality of worker threads;

5 computer-readable program code means for receiving, onto an incoming queue, a plurality  
6 of incoming client requests for connections;

7 computer-readable program code means for transferring each of said received client  
8 requests for connections from said incoming queue to a wide queue, said wide queue comprising a  
9 plurality of queues wherein each of said queues is separately synchronization-protected; and

10 computer-readable program code means for servicing, by said plurality of worker threads,  
11 said client requests by retrieving selected ones of said client requests from said queues that  
12 comprise said wide queue.

1 Claim 2 (previously presented): The computer program product according to Claim 1, wherein  
2 said computer-readable program code means for transferring further comprises:

3 computer-readable program code means for placing each of said received client requests  
4 on a selected one of said plurality of queues using a First-In, First-Out (FIFO) strategy, wherein  
5 said selected one of said plurality of queues is selected using a round-robin approach.

1 Claim 3 (previously presented): A computer program product for enhancing performance of a  
2 multithreaded application, said computer program product embodied on a computer-readable

Serial No. 09/575,938

-2-

Docket RSW9-2000-0036-US1

3 medium and comprising:

4 computer-readable program code means for executing a plurality of worker threads;

5 computer-readable program code means for receiving a plurality of incoming client  
6 requests onto a queue, wherein each of said client requests is for a connection to a host;

7 computer-readable program code means for retrieving, by an individual one of said worker  
8 threads, a selected one of said client requests from said queue;

9 computer-readable program code means for determining a number of connections to said  
10 host to which said connection is requested in said selected client request, wherein said number are  
11 those which are currently assigned to one or more of said worker threads; and

12 computer-readable program code means for processing said selected client request if said  
13 number is less than an upper limit, and for not processing said selected client request otherwise.

1 Claim 4 (original): The computer program product according to Claim 3, wherein said upper  
2 limit is a system-wide value.

1 Claim 5 (original): The computer program product according to Claim 3, wherein said upper  
2 limit is a value specific to said host to which said connection is requested.

1 Claim 6 (original): The computer program product according to Claim 5, wherein said value is  
2 dynamically computed, and further comprising:

3 computer-readable program code means for executing a supervisor thread;

4 computer-readable program code means for monitoring, by said supervisor thread,

Serial No. 09/575,938

-3-

Docket RSW9-2000-0036-US1

5 whether connections to each of said hosts succeed or fail; and  
6 computer-readable program code means for decrementing said value when said  
7 connections to said host fail.

1 Claim 7 (original): The computer program product according to Claim 6, further comprising:  
2 computer-readable program code means for incrementing said value when said  
3 connections to said host succeed.

1 Claim 8 (previously presented): The computer program product according to Claim 6, wherein  
2 said computer-readable program code means for monitoring further comprises:

3 computer-readable program code means for setting, by each of said worker threads, a  
4 thread time stamp when said worker thread performs active work for any selected one of said  
5 hosts;

6 computer-readable program code means for comparing, by said supervisor thread, said  
7 thread time stamp for each of said worker threads to a system time, thereby computing an elapsed  
8 time for said worker thread; and

9 computer-readable program code means for deactivating said worker thread and  
10 concluding that a connection to said selected host has failed if said elapsed time exceeds a  
11 maximum allowable time.

1 Claim 9 (original): The computer program product according to Claim 3, further comprising:  
2 computer-readable program code means for providing information for each of said hosts,

3 said information comprising an address of said host and a plurality of in-use flags;

4 computer-readable program code means for setting a selected one of said in-use flags  
5 when a particular worker thread is processing work on said connection to a particular host,  
6 wherein said selected one of said in-use flags is associated with said particular worker thread; and

7 computer-readable program code means for resetting said selected one of said in-use flags  
8 when said particular worker thread stops processing work on said connection to said particular  
9 host; and

10 wherein said computer-readable program code means for determining said number of  
11 currently-assigned connections further comprises computer-readable program code means for  
C 12 counting how many of said in-use flags are set.

1 Claim 10 (original): The computer program product according to Claim 3, wherein said queue is  
2 a wide queue comprised of a plurality of First-In, First-Out (FIFO) queues.

1 Claim 11 (currently amended): A system for enhancing performance of a multithreaded  
2 application, comprising:

3 means for executing a plurality of worker threads;

4 means for receiving, onto an incoming queue, a plurality of incoming client requests for  
5 connections;

6 means for transferring each of said received client requests for connections from said  
7 incoming queue to a wide queue, said wide queue comprising a plurality of queues wherein each  
8 of said queues is separately synchronization-protected; and

9 means for servicing, by said plurality of worker threads, said client requests by retrieving  
10 selected ones of said client requests from said queues that comprise said wide queue.

1 Claim 12 (previously presented): The system according to Claim 11, wherein said means for  
2 transferring further comprises:

3 means for placing each of said received client requests on a selected one of said plurality  
4 of queues using a First-In, First-Out (FIFO) strategy, wherein said selected one of said plurality of  
5 queues is selected using a round-robin approach.

C 1 Claim 13 (previously presented): A system for enhancing performance of a multithreaded  
2 application, comprising:

3 means for executing a plurality of worker threads;

4 means for receiving a plurality of incoming client requests onto a queue, wherein each of  
5 said client requests is for a connection to a host;

6 means for retrieving, by an individual one of said worker threads, a selected one of said  
7 client requests from said queue;

8 means for determining a number of connections to said host to which said connection is  
9 requested in said selected client request, wherein said number are those which are currently  
10 assigned to one or more of said worker threads; and

11 means for processing said selected client request if said number is less than an upper limit,  
12 and for not processing said selected client request otherwise.

1 Claim 14 (original): The system according to Claim 13, wherein said upper limit is a system-wide  
2 value.

1 Claim 15 (original): The system according to Claim 13, wherein said upper limit is a value  
2 specific to said host to which said connection is requested.

1 Claim 16 (original): The system according to Claim 15, wherein said value is dynamically  
2 computed, and further comprising:

3 means for executing a supervisor thread;

4 means for monitoring, by said supervisor thread, whether connections to each of said hosts  
5 succeed or fail; and

6 means for decrementing said value when said connections to said host fail.

1 Claim 17 (original): The system according to Claim 16, further comprising:

2 means for incrementing said value when said connections to said host succeed.

1 Claim 18 (previously presented): The system according to Claim 16, wherein said means for  
2 monitoring further comprises:

3 means for setting, by each of said worker threads, a thread time stamp when said worker  
4 thread performs active work for any selected one of said hosts;

5 means for comparing, by said supervisor thread, said thread time stamp for each of said  
6 worker threads to a system time, thereby computing an elapsed time for said worker thread; and

Serial No. 09/575,938

-7-

Docket RSW9-2000-0036-US1

7 means for deactivating said worker thread and concluding that a connection to said  
8 selected host has failed if said elapsed time exceeds a maximum allowable time.

1 Claim 19 (original): The system according to Claim 13, further comprising:

2 means for providing information for each of said hosts, said information comprising an  
3 address of said host and a plurality of in-use flags;

4 means for setting a selected one of said in-use flags when a particular worker thread is  
5 processing work on said connection to a particular host, wherein said selected one of said in-use  
6 flags is associated with said particular worker thread; and

7 means for resetting said selected one of said in-use flags when said particular worker  
8 thread stops processing work on said connection to said particular host; and

9 wherein said means for determining said number of currently-assigned connections further  
10 comprises means for counting how many of said in-use flags are set.

1 Claim 20 (original): The system according to Claim 13, wherein said queue is a wide queue  
2 comprised of a plurality of First-In, First-Out (FIFO) queues.

1 Claim 21 (currently amended): A method for enhancing performance of a multithreaded  
2 application, comprising the steps of:

3 executing a plurality of worker threads;

4 receiving, onto an incoming queue, a plurality of incoming client requests for connections;

5 transferring each of said received client requests for connections from said incoming queue

6 to a wide queue, said wide queue comprising a plurality of queues wherein each of said queues is  
7 separately synchronization-protected; and  
8 servicing, by said plurality of worker threads, said client requests by retrieving selected  
9 ones of said client requests from said queues that comprise said wide queue.

1 Claim 22 (previously presented): The method according to Claim 21, wherein said transferring  
2 step further comprises the step of:

3 placing each of said received client requests on a selected one of said plurality of queues  
4 using a First-In, First-Out (FIFO) strategy, wherein said selected one of said plurality of queues is  
5 selected using a round-robin approach.

C  
1 Claim 23 (previously presented): A method for enhancing performance of a multithreaded  
2 application, comprising the steps of:

3 executing a plurality of worker threads;

4 receiving a plurality of incoming client requests onto a queue, wherein each of said client  
5 requests is for a connection to a host;

6 retrieving, by an individual one of said worker threads, a selected one of said client  
7 requests from said queue;

8 determining a number of connections to said host to which said connection is requested in  
9 said selected client request, wherein said number are those which are currently assigned to one or  
10 more of said worker threads; and

11 processing said selected client request if said number is less than an upper limit, and not



12 processing said selected client request otherwise.

1 Claim 24 (original): The method according to Claim 23, wherein said upper limit is a system-wide  
2 value.

1 Claim 25 (original): The method according to Claim 23, wherein said upper limit is a value  
2 specific to said host to which said connection is requested.

C  
1 Claim 26 (original): The method according to Claim 25, wherein said value is dynamically  
2 computed, and further comprising the steps of:  
3 executing a supervisor thread;  
4 monitoring, by said supervisor thread, whether connections to each of said hosts succeed  
5 or fail; and  
6 decrementing said value when said connections to said host fail.

1 Claim 27 (original): The method according to Claim 26, further comprising the step of  
2 incrementing said value when said connections to said host succeed.

1 Claim 28 (previously presented): The method according to Claim 26, wherein said monitoring  
2 step further comprises the steps of:  
3 setting, by each of said worker threads, a thread time stamp when said worker thread  
4 performs active work for any selected one of said hosts;

Serial No. 09/575,938

-10-

Docket RSW9-2000-0036-US1

5 comparing, by said supervisor thread, said thread time stamp for each of said worker  
6 threads to a system time, thereby computing an elapsed time for said worker thread; and  
7 deactivating said worker thread and concluding that a connection to said selected host has  
8 failed if said elapsed time exceeds a maximum allowable time.

1 Claim 29 (original): The method according to Claim 23, further comprising the steps of:

2 providing information for each of said hosts, said information comprising an address of  
3 said host and a plurality of in-use flags;

4 setting a selected one of said in-use flags when a particular worker thread is processing  
5 work on said connection to a particular host, wherein said selected one of said in-use flags is  
6 associated with said particular worker thread; and

7 resetting said selected one of said in-use flags when said particular worker thread stops  
8 processing work on said connection to said particular host; and

9 wherein said step of determining said number of currently-assigned connections further  
10 comprises counting how many of said in-use flags are set.

1 Claim 30 (original): The method according to Claim 23, wherein said queue is a wide queue  
2 comprised of a plurality of First-In, First-Out (FIFO) queues.

1 Claim 31 (currently amended): The computer program product according to Claim 1, further  
2 comprising:

3 computer-readable program code means for returning each of said retrieved selected ones

Serial No. 09/575,938

-11-

Docket RSW9-2000-0036-US1

4 of said client requests for which work has not yet completed to a selected one of said plurality of  
5 queues in said wide queue, wherein said selected one of said plurality of queues is selected using  
6 queue using a round-robin approach, upon approach upon completion of said computer-readable  
7 program code means for servicing.

1 Claim 32 (currently amended): The system according to Claim 11, further comprising:

2 means for returning each of said retrieved selected ones of said client requests for which  
3 work has not yet completed to a selected one of said plurality of queues in said wide queue,  
4 wherein said selected one of said plurality of queues is selected using queue using a round-robin  
5 approach, upon approach upon completion of said means for servicing.

1 Claim 33 (currently amended): The method according to Claim 21, further comprising the step  
2 of:

3 returning each of said retrieved selected ones of said client requests for which work has  
4 not yet completed to a selected one of said plurality of queues in said wide queue, wherein said  
5 selected one of said plurality of queues is selected using queue using a round-robin approach,  
6 upon approach upon completion of said servicing step.